We Need to Discuss the Relationship
An Analysis of Facilitators and Barriers of Software Ecosystem Partnerships

George Valença[1] and Carina Alves[2]
[1]*Departamento de Estatística e Informática, Universidade Federal Rural de Pernambuco, Recife, Brazil*
[2]*Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil*

Keywords: Software Ecosystems, Partnerships, System Dynamics, Strategies.

Abstract: Software ecosystems are a promising paradigm to develop and market software systems by means of partnerships among companies. To ensure the healthy evolution of software ecosystems, companies must define strategies that strength their partnerships. In this paper, we investigate the factors that drive the evolution of software ecosystems formed by Small-to-Medium Enterprises. We present an exploratory case study of two emergent software ecosystems in order to analyse the main facilitators and barriers faced by participating companies. We adopt System Dynamics approach to create models expressing causal relations among these factors. By understanding the facilitators that should be reinforced and barriers that should be restrained, we believe that partners are better equipped to catalyse the success of their software ecosystems.

1 INTRODUCTION

Software ecosystems figure among the most recent and relevant trends in IT industry. A software ecosystem is as a set of businesses functioning as a unit and interacting with a shared market for software and services, together with the relationships among them (Jansen et al., 2009). They involve the interdependence and interrelation to external partners and stakeholders with which a software company collaborates and competes (Olsson and Bosch, 2016). Software ecosystems promote the idea of coopetition, when companies embrace competitive collaborations and start to co-evolve their products in a hub of local and/or global market (Popp, 2013). By defining partnerships to engage in this networked setting, companies acquire new skills, share features and clients, and divide R&D costs (Bosch, 2009). Moreover, they can cope with financial, time and knowledge constraints (Khalil et al., 2011). Successful examples of software ecosystems include Apple's iPhone and the range of complementary apps developed by third-party players available at Apple Store, Eclipse open source ecosystem, among other platforms available in the IT industry. The increasing growth of software ecosystems confirms that companies co-existing in the same market have recognised their need to cooperate to survive in a turbulent environment.

This paper reports on an exploratory case study of two emergent software ecosystems formed by Small-to-Medium Enterprises (SMEs). The tight relationships among these companies result from frequent joint projects to integrate their ERP software solutions and services. On the one hand, SMEs must cope with limited financial and human resources. On the other hand, they have flexible organisational structure and motivation to explore innovative business models. These aspects direct the way SMEs define partnerships and position themselves in a software ecosystem.

The motivation to conduct this research is to investigate the factors that affect positively and/or negatively the evolution of partnerships among SMEs establishing a software ecosystem. We achieved this goal by adopting System Dynamics method (Senge and Kurpius, 1993) to analyse the factors that nurture and/or hamper the partnerships. The contribution of our study lies in describing these factors and presenting diagrams expressing causal relations among them. Besides, we present strategies that enable software companies to understand what drives the healthy evolution of their ecosystems.

This paper is organised as follows. Section 2 describes the conceptual background of the research. Section 3 details the research method. Section 4 presents systemic diagrams by adopting System Dynamics. Section 5 uses the diagnostic of the

Valença, G. and Alves, C.
We Need to Discuss the Relationship - An Analysis of Facilitators and Barriers of Software Ecosystem Partnerships.
DOI: 10.5220/0006231900170028
In *Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 2*, pages 17-28
ISBN: 978-989-758-248-6

partnerships to delineate strategies that companies can adopt to foster the evolution of the software ecosystems, in light of literature in the field. Finally, Section 6 presents final considerations.

2 CONCEPTUAL BACKGROUND

2.1 Software Ecosystems

Partnerships differ from more general business relationships due to (i) firms' degree of mutual commitment, respect, trust and influence; (ii) communication behaviour that involves transparency and information sharing; and (iii) tendency towards joint problem solving; among other properties (Mohr and Spekman, 1994). They are the seed of a software ecosystem by allowing external actors to customise or complement the features of existing products, and provide technical services (Cusumano, 2004).

This network changes the dominant logic of doing business, based on integrated manufacturing, in-house R&D and direct sales. Ecosystem partners focus on innovative business models, which involve novel ways for a firm to collaborate with external agents and for them to create and capture value from the network (Weiblen, 2015).

The actors of a software ecosystem have different roles and responsibilities. Manikas and Hansen (2013) provide an overview of the most common actors in a software ecosystem, which includes keystone, niche player, external developer or third party developers, vendor or reseller, customer and user. The keystone has a critical function, since it guarantees the well-functioning of the ecosystem. This player is responsible for running the software platform, creating and applying rules, processes and business procedures, setting and monitoring quality standards, and orchestrating actors' relationships. Niche players are also central to the ecosystem, as companies that use the platform to develop or add components (e.g. apps) to it, producing functionality that customers require. They create or enhance capabilities that differentiate them from other participants. Their importance lies in complementing keystone work and influencing decision-making in ecosystem management.

All actors are committed to a certain degree to ensure their own health as well as their partners' health in the ecosystem. Hence, ecosystem prosperity represents their own prosperity. Hartigh and colleagues (2006) argue the health of an ecosystem is a way of assessing its strength at a specific moment. Iansiti and Levien (2004) propose a classification inspired on biological ecosystems to define health as the extent to which an ecosystem as a whole is durably creating opportunities for its members and those who depend on it. The three measures of health are *productivity*, *robustness* and *niche creation.* *Productivity* indicates the ecosystem ability to transform inputs into products and services. Number of applications in an App Store is a possible way of measuring the productivity of a software ecosystem. *Robustness* indicates the ecosystem capacity to deal with interferences and competition pressure. The survival rate of ecosystem members is a possible metric to assess this aspect. Finally, *niche creation* represents the opportunities for actors available in the ecosystem. It fosters diversity by creating valuable resources and niches. The number of new players around the platform is a way to assess niche creation.

2.2 System Dynamics

System Dynamics (SD) provides understanding about the structure and functioning of systems in which we are embedded. The approach supports the definition of high-leverage policies for sustained improvement. System behaviour is represented by graphical schemes that combine reinforcing and balancing cycles formed by variables from studied phenomena. Reinforcing loops are the engine of growth and can be virtuous (situations that reinforce in desired directions) or vicious (situations that start badly and grow worse). Balancing loops maintain the status quo of a given context. Many loops also contain delays, which highlight consequences (i.e. factor *x* foster factor *y*) that will gradually occur (Senge and Kurpius, 1993).

These schemes can be associated with one or more of the 13 existing generic system archetypes. Each archetype has a script that guides the interpretation of the investigated context (Senge and Kurpius, 1993). The selection of an archetype depends on how the related script properly describes the studied phenomenon. This is done by identifying contextual variables that hold cause and effect relations that fit the archetype script. The use of system archetypes is a rich technique to describe or predict the behaviour of a system by drawing related causal loops of variables from the studied scenario. Hence, it is possible to either analyse a past situation or forecast specific scenarios by identifying potential traps and mitigating risks of occurrence. We highlight that the effectiveness of SD approach depends on the ability of researchers to reflect and comprehend the reality under study.

3 RESEARCH METHOD

Our multiple case study analysed the main drivers of partnerships among Small-to-Medium Enterprises participating in a software ecosystem. We translate this goal in the following research questions (RQ):

- RQ1 – What are the facilitators and barriers of partnerships among software companies participating in a software ecosystem?
- RQ2 – How the facilitators and barriers factors interact with each other?
- RQ3 – What strategies can leverage the success of the software ecosystem in light of these factors?

To address these research questions, we performed 2 case studies (Case Study I – CSI and Case Study II – CSII) composed of 5 and 3 software companies, respectively. We purposefully selected them in order to obtain information-rich cases to investigate the phenomenon of software ecosystem partnerships in depth (Coyne, 1997).

3.1 Case Companies

CSI involved 5 partner companies from Recife, Brazil, here named as Company A, Company B, Company C, Company D and Company E (Table 1). The companies integrate their products in frequent joint projects, which are started by one or more partners. By strengthening their relationships, the partners have gradually created a software ecosystem formed by the integrated software systems developed by these complementary companies. We initiated our study analysing the partnership between Company A and Company B. Preliminary interviews enabled us to identify other relevant players: Company C, which is partner of Company A and Company B; and Company D, which is partner of Company B. We then mapped Company E, as partner of Company D.

Table 1: Overview of Companies A, B, C, D and E.

Company	Solutions
Company A	ERP with 5 modules focused on retail chains, distributors and wholesalers markets
Company B	ERP with 15 modules focused on several market niches (e.g. healthcare, oil and gas, sugar industry and logistics)
Company C	Information system with 3 modules for pharmacies
Company D	10 information systems for hospitals
Company E	Web portal for electronic quotations

CSII involved 3 software firms operating in Recife and São Paulo, Brazil, here named as Company F, Company G and Company H (Table 2). They build a software ecosystem in which Company F is the keystone and the main responsible for sharing business deals with partners.

Table 2: Overview of Companies F, G and H.

Company	Solutions
Company F	ERP solutions with 60 modules for hospitals and healthcare market
Company G	20 information systems for laboratories
Company H	Services to revamp software systems in diverse markets

We started CSII by exploring the partnership between Companies F and G, which involves the integration of complementary healthcare solutions. In addition, we mapped the partnership between Company F and Company H, which is critical to maintain the products of Company F.

3.2 Data Collection

We undertook open-ended and semi-structured interviews to map the factors that enable and inhibit a partnership in a software ecosystem, which we name as *facilitators* and *barriers*. We interviewed 20 professionals in CSI (Table 3) and 7 professionals in CSII (Table 4). The participants played both technical and managerial roles.

Table 3: Interviewees from Companies A, B, C, D and E.

Company	Function
Company A	Project Manager, Business Analyst, System Analyst.
Company B	Project Manager, Product Manager, Release Manager, Integration Team Leader, Business Analyst, System Analyst (2), Tester.
Company C	Services Manager, Project Manager, Business Analyst.
Company D	Product Manager, Project Manager, Solutions Architect, System Architect, System Analyst.
Company E	Operations and Deployment Director.

Table 4: Interviewees from Companies F, G and H.

Company	Function
Company F	Sales Director, Marketing Manager, Product Owner, Business Analyst, System Analyst.
Company G	Marketing Manager.
Company H	Operations Director.

One author conducted and transcribed the 27 interviews. The transcripts were later analysed with the other author to reach an agreed understanding about the collected data and discuss the findings.

3.3 Data Analysis

We started data analysis by searching for barriers and facilitators in interviews discourse. Then, we mapped factors that were common to both cases, as a means to represent key drivers of partnerships. These factors are considered as variables in System Dynamics method (Senge and Kurpius, 1993). We listed and crossed them in a table to examine causal relations among them. Once we identified a possible relation in such causal matrix, we inserted a code *d* or *i* to indicate that the variable in the line caused the variable in the column in a directly (*d*) or inversely (*i*) proportional form, respectively. We also labelled each relation with the values *1* and *3* to indicate standard weights related to causal relations intensity. We crossed the factors considering interviews evidence and our interpretation of facts.

Then, we created SD models to represent the variables and correspondent relations. We considered the most relevant variables (i.e. with greater systemic power in the matrix). By selecting variables with high values of influence, we also avoided the complexity explosion that would result from a large number of contextual variables and relationships in the models. The subsequent step was the identification of a subset of variables considered as critical, based on our interpretation and interviewees' opinions. The resultant model presents the barriers and facilitators to describe the dynamics of the studied context in a graphical form. It denotes leverage points and causal cycles that contribute to or limit the healthy evolution of the ecosystem.

In a final step, we discussed the SD models in evaluation interviews with the studied companies. During this process, we asked participants (i) whether the diagram represented the appropriate elements (factors and relations), and (ii) whether there were other elements to include. As a result, we performed some punctual refinements in SD models.

4 SYSTEMIC VIEW OF FACILITATORS AND BARRIERS IN PARTNERSHIPS

In Section 4.1, we present the facilitators and barriers that influence the partnerships among companies of CSI and CSII. Section 4.2 describes the SD models generated for our multiple case study.

The models present a synthesised view of facilitators and barriers identified in the partnerships of studied companies. Given the fact that companies of CSI and CSII share similar contextual factors (e.g. size, geographical location, ERP application domain, types of partnerships), we opted to conduct an integrated analysis of facilitators, barriers and the resulting systemic archetypes of CSI and CSII.

4.1 Facilitators and Barriers

This section answers RQ1 by describing a set of facilitators and barriers for the studied companies to thrive in their software ecosystems. Facilitators (F) are factors that can contribute to the creation and growth of partnerships. Our analysis of CSI and CSII revealed the following seven facilitators:

- **F1 – Personal and Geographical Proximity**

Companies' physical proximity promotes joint projects among them: *"since it (Company B) was near, we took the software from Company B"*, cited the software architect from Company D. In particular, companies that operate in the same region understand the specific needs of this market. Hence, geographically and personally close firms often become relevant partners.

The joint projects start with strict professional relationships among staff of partners companies (e.g. managers in an integration project). Once these interactions evolve to more personal relationships, the companies can benefit from a good communication channel and professionals that aim to leverage the partnership. The arguments of the marketing manager from Company G illustrate this scenario: *"since we have worked very well (and) I already know several people from Company F, we will try to grow this partnership; it is a communication channel that facilitates a lot; the partnership flows very well"*.

It means that the closer personal relationship between staff of partner companies catalyses their collaboration, as the services manager from Company C detailed: *"I 'hit the door' of Company A to talk to the president to seek business opportunities"*. Their personal relations facilitate the execution of projects: *"communication (with Company B) flows well since this team worked together in other projects; it emerged a friendship outside the firm; this helps a lot"*, argued the project manager from Company A.

- **F2 – Respectful Attitude**

Companies that keep a respectful attitude are fostering their partnership. The marketing manager

from Company G highlighted the relevance of a good conduct: *"these are companies that always respected each other, which is very important; Company F never mentioned anything (related to paying) a commission; they are very professional and Company G never offered anything to them"*. The companies appreciate such attitude of a partner, which increases the trust on the other company.

- **F3 – Mutual Trust**

Trust results from the close relationship of the firms, sometimes a personal proximity of their members. It is seen as a premise for a partnership to emerge, as cited by the innovation manager from Company C: *"to establish a partnership, you already trust the partner; you already have confidence, you know he is responsible; it is a premise for you to establish a partnership"*.

The commercial director from Company F also argued the dependence kept by partners requires full reliability: *"(we must have) total trust, because they hold key knowledge of the code"*. The trust factor increases with actions that favour joint projects, such as promptly treating systems integration problems. The project manager from Company A clarified the relevance of trust: *"I'd rather have a less competent but reliable partner than a super competent but unreliable one"*.

- **F4 – Openness for Technical and Business Negotiation**

Flexibility for business and technical negotiations is a critical factor for a partnership to succeed, because companies must guarantee a win-win approach. Companies mentioned positive experiences with partners who were open to discuss technical and business issues. The marketing manager from Company G discussed this fact: *"there is this technical part, when we can integrate (our systems) very well; people get (access to) the necessary channels, (where) people are open and available to help us at any time"*.

The project manager from Company D exemplified the impact of this factor in a relationship: *"the partner approached us with interest and humility; (another) partnership did not evolve because the partner was inflexible"*. The commercial director from Company F also stressed the importance of easily negotiating commercial issues with partners: *"any integration that I do consumes time with maintenance, installation, or a failure; (so) it is fair that part of it (payment) is reverted to me; there are firms that are very open; others (are) inflexible; this is very bad"*.

Openness for negotiation is a common trend among SMEs: *"when there is the possibility to negotiate is because they are firms of the same size; in general, there is (openness for negotiation)"*, declared the operations director from Company B.

- **F5 – Effective Communication**

Good communication is essential for an integration project to succeed and thereby for a partnership to evolve. It is important to establish adequate communication channels and ensure the right people are available to have technical or business discussions with a partner. For instance, Company F maintains an integration team, which has a wider view of the integrations between its healthcare solutions and other systems. The marketing manager from Company G explains such facilitator: *"we have much trouble to get to the person who will develop the integration; this is something that really makes it (integration) difficult; the (partnership of) Company G with Company F works because of the right channels; it is the best (communication channel) we have (with a partner)"*.

- **F6 – Perceived Quality of Products and Services**

Quality of products and services is a criterion considered by companies to select a partner, given the relevance of quality for client satisfaction. For instance, Company D considers the quality of a partner team and services as a premise to establish a partnership. Its project manager illustrated this situation: *"in addition to off the field factors like 'whether a partner has a qualified team, (we analyse) whether his services desk is good'"*.

The companies assess the quality of a system from another vendor as an indicator to invest in a new partnership. *"A partner would hardly be invited if beforehand we knew that he would not satisfy (our quality criteria)"*, cited Company F business analyst. The marketing manager from Company G explained the relevance of this factor: *"they recommend us because they know they will not have problems in the integration; they will recommend a firm to stay in their client; if it is a bad firm, which gives many problems, the system crashes; it is worse for them"*. Low quality solutions can affect the reputation of a company as a supplier, as described by the commercial director from Company F: *"the quality of the product, its acceptance in the market and how much it adds value (to mine)…; we need to choose well our partners since (they) will, in a way, influence our (system) routines and reputation"*.

- **F7 – Availability of Standards/Technologies to Support Systems Integration**

By defining or adopting integration standards, partners facilitate their collaboration, as illustrated by the software architect from Company D: *"the 'Integrator' has helped and now HL7 (international standards for data transfer between healthcare*

systems) will help even more". A common integration infrastructure reduces mismatches among products and rework in joint projects. *"One of the main gains we will have (with integration technology) is to prevent us from redeveloping integrations whenever someone knocks our door"*, argued the project manager from Company D. One partner may develop the integration infrastructure or it may emerge as a joint creation: *"we are aligning what one has with what the other has, what one may add... Company C can contribute with definitions, Company B with human resources, etc."*, explained the release manager from Company B.

The following paragraphs present nine Barriers (B) of studied partnerships. These factors are the opposite view of facilitators: they weaken companies' relationships and disturb their joint projects. Therefore, barriers may reduce the health of their software ecosystems.

- **B1 – Inefficiency to Handle Integration Problems**

The non-involvement of a firm in the analysis of system integration problems strongly weakens a partnership: *"(another) partnership did not evolve (because) the partner was uncompromising and did not resolve (the issue)"*, explained the project manager from Company D. Given the fuzzy boundary among integrated systems, firms may try to convince the partner that the problem is originated in his system: *"the partner often shifts the issue to the other (company)"*, declared the team leader from Company B. Others simply prioritise other projects, as reported by the project manager from Company A: *"sometimes the partner has more critical issues in another project and leaves (ours) behind"*. This is common in the context of SMEs, which are often overwhelmed, handling demands of multiple clients with limited resources. Such low attention may also happen because the partner does not see the client as strategic: *"sometimes partners do not give attention since (the client) is not in their top customer base"*, cited the services manager from Company C.

When the client is not aware that multiple vendors are providing the solution or simply does not understand their duties in a joint project, it is hard to know who to blame and appeal. Handling integration issues demands a clear definition of roles and responsibilities among partners. However, a partner may refuse such managerial responsibility.

To avoid client dissatisfaction, some firms take the duties of a partner not to jeopardise their reputation. Company F currently treats this issue by managing customer support, as described by the commercial director: *"we concentrate the support within our firm and meet specific demands by contacting the customers"*. The lack of support reduces companies' trust in a partner, who is no longer recommended to clients. *"Our manager asks not to contact Company B; we try to solve the issue here; nowadays we do not recommend Company B"*, cited the systems analyst from Company D.

- **B2 – Unavailability of a Professional to Manage Systems Integration**

The absence of a permanent employee responsible for the integrated solution is a problem, as described by the product manager from Company B: *"I change part of the process, but this brings a big risk, because you do not have someone in charge of the whole (integration)"*. Defining a professional as the 'owner of the integration' facilitate negotiations and alignment of products, according to the services manager from Company C: *"we do not have this guy, which would be the focal point; such confusion and discussions would be minimised; (he would be in charge of) communication and sharing of information"*. The duties of an integration owner include the identification of evolution needs due to market demands and analysis of impact of product changes in integrations. However, they cannot afford his salary without a running project: *"There must be someone paying him, (but) we are project-oriented"*, cited the services manager from Company C.

- **B3 – Weak Commercial/Prices Alignment**

The commercial alignment of the firms is critical for a partnership to evolve: *"this is crucial, because if partner price is not feasible for the deal, we have to look for another (product)"*, argued the operations director from Company B. Partners who define high price weaken negotiations with clients. *"(Product) price affect negotiations; we have to talk to partner board to lower costs; it hinders some partnerships"*, argued Company E's operations and deployment director. This attitude leads to gradual replacement of these SMEs in the ecosystem or the development of the complementary system by the other company: *"we normally sell with the software from Company B, but their cost may turn the proposal expensive; if we had a financial module, it would cheapen the (final) system"*, explained the system analyst from Company D. The high prices asked by Company G might derail their partnership. *"We are pressuring Company G to lower prices because they are moving the market away; if we do not reach a consensus, I would opt for another partner"*, cited him.

Similarly, a vendor from Chile required a high price to include its system in a joint project, which increased the final price of the proposal and made Company F rethink this partnership. *"I chose to use an ERP that was already adapted to Chile, (but) it made my offer very costly; we negotiate, but it is hard"*, argued the commercial director. Due to the partner's size and position in the Chilean market,

Company F was dependent on its system. Therefore, the partner had bargaining power. *"This is a world-class player, much bigger than us; we can have a policy to reduce (our prices), but they (decide) in this case; they said 'my price is this'; it is always better (to align with smaller partners)"*, added him.

- **B4 – Poor Strategic Alignment of Products**

Strategic alignment of products is necessary for partnerships to thrive. *"It is very difficult to reconcile strategies and portfolios; but it is also very difficult to survive without (it)"*, declared the product manager from Company D. The project manager from Company D reinforced this fact: *"lack of synchronisation is very negative"*. As the scope of systems integration grows, the dependence between products increases, as highlighted by the system analyst from Company B: *"the conflict is: I'm evolving and I can damage something there or there may be a need and we are eliminating for disuse"*.

So far, studied firms have not formally aligned product strategies (e.g. roadmaps, releases) and are not prepared to jointly evolve the systems: *"one thing that makes the partnership fragile is that when I do my strategic planning, I do it independently of them"*, cited Company B operations director. The challenge to ensure such alignment stem from the fact that firms attempt to manage partnerships and parallel demands with restricted resources: *"it is difficult to (align releases) since the partner has things going on outside partnerships"*, explained the innovation manager from Company C.

The well-functioning of a product integration must be ensured during the lifetime of the systems from different vendors, which demands a technical and strategic alignment. However, such convergence may not be perceived in partners' daily operation, as argued by the commercial director from Company F: *"in all partnerships / integrations we had, there is a great and natural difficulty: I have a product evolving with a great speed and the partner cannot follow it"*. Product strategic alignment is essential in software ecosystems, when there is a great mutual dependence between firms. *"In a simple integration, I do not have to make roadmap alignment with him (partner); with Company H, which has a framework that needs to evolve over time, if we do not evolve together, we go anywhere; some partnerships are for survival"*, argued him. The same occurs for the technologies: *"another issue is the technological misalignment between products; when technology is adherent (it is easy to integrate)"*.

- **B5 – Overlap between Features Offered by the Company and Potential Partners**

Studied firms prioritise fully complementary vendors; as such partnership does not require scope negotiation (i.e. decide which feature will be provided, in case it is available in both systems). The commercial director from Company F explained this fact: *"(a problem emerges) when there is a conflict of interests; when (partner's) product is competing with our (solution); it is forbidden to do this"*. If a vendor offers a system in the same domain of interest of the firm, it has reduced chances to become a partner. In this case, there may be punctual collaborations: *"in some situations we can even establish an isolated partnership, but we do not define the partnership in a fixed form (which) can be used in any (client) environment"*, cited him.

Company F aims to keep its independence on partners in specific areas: *"some areas are reserved for us; we do not facilitate"*. When the area is strategic, the company develops the feature instead of searching for a partner. This attitude stems from the possibility that a partner has to decrease the envisaged market share of Company F, as he argued: *"the main restriction (for a partnership) is if that (system) enters an area in which I offer (solutions) or want to offer; the company kills the possibility that we have to grow in this market"*.

- **B6 – Differences in Working Practices**

In our case studies, some partners have very different working practices: *"we have releases almost every fortnight; the partner says 'I can only (deliver it) in 3 months'; it is another pace, another process"*, detailed the project manager from Company A. Such differences pose challenges to joint development, as the services manager from Company C explained: *"if I tell him (partner client) I deliver it (feature) in 6 months, he says it makes sense; if I tell (it to) him (retailer client), he send me away"*. This situation makes Company C to have slower deliveries with Company B: *"it has to do with culture; this was the difficulty with Company B; all happened very methodically"*, added the services manager. Even the evolution of technologies can be harmed, as the operations and deployment director from Company E described: *"sometimes we are well ahead of partners; we want them to evolve and sometimes (their processes) are too rigid"*.

- **B7 – Limited Authority Over Partner's Development Team**

The lack of authority of a firm over the partner's team is an issue faced by firms in joint projects. *"You have to manage a team (in charge of) another system, from another firm over which you have no authority; it is too complicated"*, argued the project manager from Company B. The project manager from Company A detailed this fact: *"when you take on this role (project manager), it has the dependence (on partners since) you do not have all that strength*

in other (software) factories". Partners can restrict the access to their teams, even when they have collaborative projects. In general, the teams only follow orders from their own firms, creating a weak functional matrix: even when a project manager leads the joint project, he must negotiate with managers from the partners to forward demands. *"I have power over my (development) factory, but I cannot impose (anything) in the factory of Company B"*, detailed the project manager from Company A.

- **B8 – Low Availability of Financial and Human Resources**

A firm with restricted financial and human resources may hamper the development of partnerships, as it involves several operational costs (e.g. travels) and strategic investments (e.g. innovation projects). In such cases, a firm is seen as unprepared for the collaboration, according to the commercial director from Company F: *"if the partner has no capacity to invest in systems integrations and products, it compromises (the relationship) as he (partner) cannot work with you"*. As small-to-medium sized firms, partners usually face big restrictions of financial and human resources, affecting joint projects. *"It (Company B) suffers from lack of resources and I cannot move on"*, argued Company A's project manager. Firms that can invest in joint projects support partnerships.

- **B9 – Short Expertise in integration projects**

Although systems integration seems a common practice for studied companies, in some situations they lack such expertise, which may harm a new partnership. The business analyst from Company F explained this fact: *"the firm with which we will make the integration may already have the (integration) know-how, the experience of doing this, which we (may) not have, maybe not at the same level"*. Inexperienced and immature companies may affect the success of integration projects.

4.2 System Dynamics Models

This section answers RQ2 by presenting SD models that analyse the causal relations among the previous facilitators and barriers. Figure 1 shows a SD model for the ecosystems from CSI and CSII. It consists of a network of causal relations among 6 facilitators (blue) and 6 barriers (red). It is important to notice that we neutralised their names by eliminating adjectives and/or adjusting the nouns. For instance, we altered the barrier *inefficiency to handle*

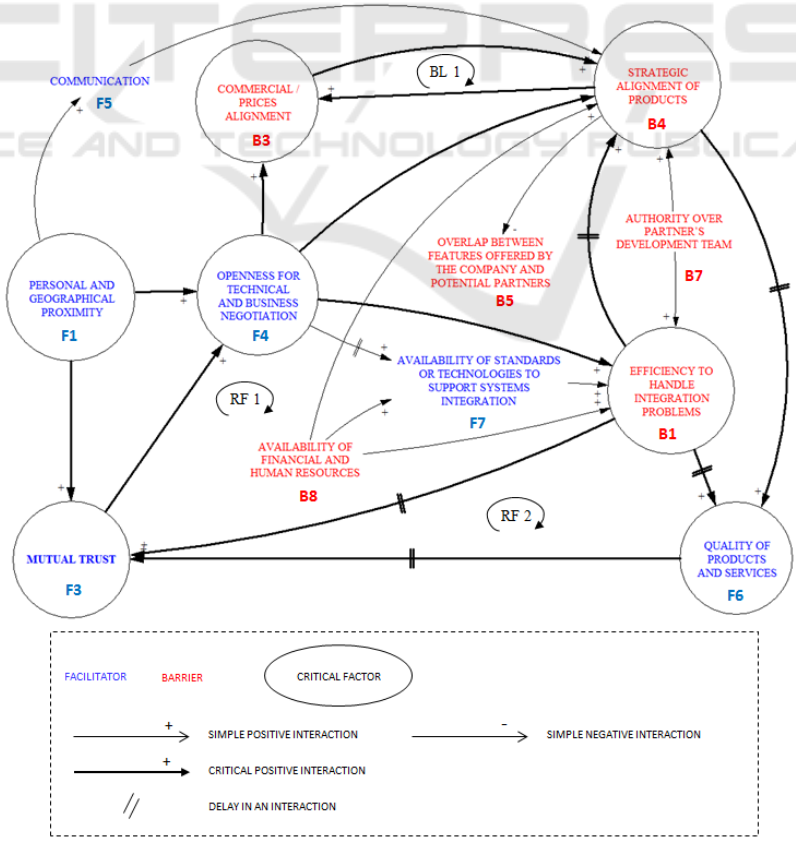

Figure 1: SD model representing interactions among barriers and facilitators.

integration problems to *effectiveness to handle integration problems*, removing its negative form.

The colours of the factors indicate how they are perceived in the studied context. This was a means to simplify the analysis of causal relations and avoid inappropriate logical comparisons. The model represents factors that already exist (e.g. *perceived quality of products and services*) and those that lack in practice (e.g. *commercial/prices alignment*). The arrows associating the factors indicate the influence they may have on each other: the factor from which an arrow leaves tends to promote the one in which the arrow arrives. For instance, *commercial/prices alignment* promotes *strategic alignment of products strategies and technologies*. However, since both are in red, one shall interpret it as *weak commercial/prices alignment* reinforces the *poor* .

We highlight the most critical factors in circles (relationships among them are also detached with thick arrows), i.e. *commercial/prices alignment*, *effectiveness to handle systems integration issues*, *personal and geographical proximity*, *strategic alignment of products*, *openness for technical and business negotiation*, *perceived quality of products and services*, and *mutual trust*. These factors were obtained from interview evidence, such as the arguments of the project manager from Company A about a partner inefficiency to handle integration problems: *"this (occurrence of issues in the integration) happens a lot; it is the biggest difficulty when we have a partner"*. The commercial director from Company F also confirmed this fact: *"this (inefficiency to treat integration problems) is an important challenge"*. Another example lies in the opinion of the operations director from Company B regarding the poor alignment of prices among partners: *"this (lack of commercial alignment) happens and we have to negotiate before (presenting a proposal to the client); because if it is not feasible we have to search for another solution; this is vital for a partnership"*.

From the SD model in Figure 1, we can perceive the virtuous reinforcing loop *RF 1*, which leads partners to effectively perform the joint projects. The *openness for technical and business negotiations* favours the *availability of standards or technologies to support systems integration* among partners, this in turn can contribute to partners' *effectiveness to handle systems integration problems*. This factor will further leverage their *mutual trust* and facilitate future negotiations. A wider view of this cycle is the virtuous reinforcing loop *RF 2*, which includes the factor *perceived quality of products and services* by clients and partners. This factor results from partners' *effectiveness to handle systems integration problems* and fosters their *mutual trust*. However, results from the simple balancing cycle *BL 1* may affect *RF 2*: the weak *commercial/prices alignment* reduces the already weak *strategic alignment of products*.

In Figure 2, we describe another representation of the barriers and facilitators. This specific view translates a system archetype called Accidental Adversaries (AA). The AA illustrates a situation in which two actors start a relationship aiming at capitalising their power and reducing their weaknesses. It is based on the idea of a healthy collaborative environment that supports a goal that cannot be achieved by parties individually. However, issues arise when one or both parties take actions they consider reasonable but that end up supressing partner's success. These harmful actions foster a sense of antipathy and may even turn partners into adversaries. This scheme synthesises some challenges that partners face in the studied software ecosystems.

In the AA archetype presented in Figure 2, the names of the factors were adjusted to represent the systemic action between two given partners in the ecosystem. We also created four variables (in grey) that were inferred from the situation at hand. In short, the outermost virtuous reinforcing cycle *RF 1* is a virtuous loop that promotes the evolution of partnerships. In their turn, the virtuous reinforcing cycles *RF 2* and *RF 3* mean individualistic actions that bring unintentional consequences that ultimately create the balancing cycles *BL 1* and *BL 2*. These balancing loops hold back the virtuous cycle of the partnerships (*RF 1*). Hence, they represent negative situations that restrict partnerships prosperity.

The former diagrams show that partners must be open for negotiations. The separate price policy of the firms is a barrier, as it hampers a partner to close a deal. By fostering their *commercial alignment*, partners enable the recurrence of joint projects. This shall increase companies' confidence in partnerships prosperity. It is then likely to observe an increase in the *strategic alignment of products*, which may ultimately promote the *quality of products and services* offered to clients. Hence, partners leverage ecosystem health and each other success as vendors.

5 DISCUSSION

Based on the former SD models and considering guidelines from the literature, we derive strategies to strengthen the collaboration among partners and

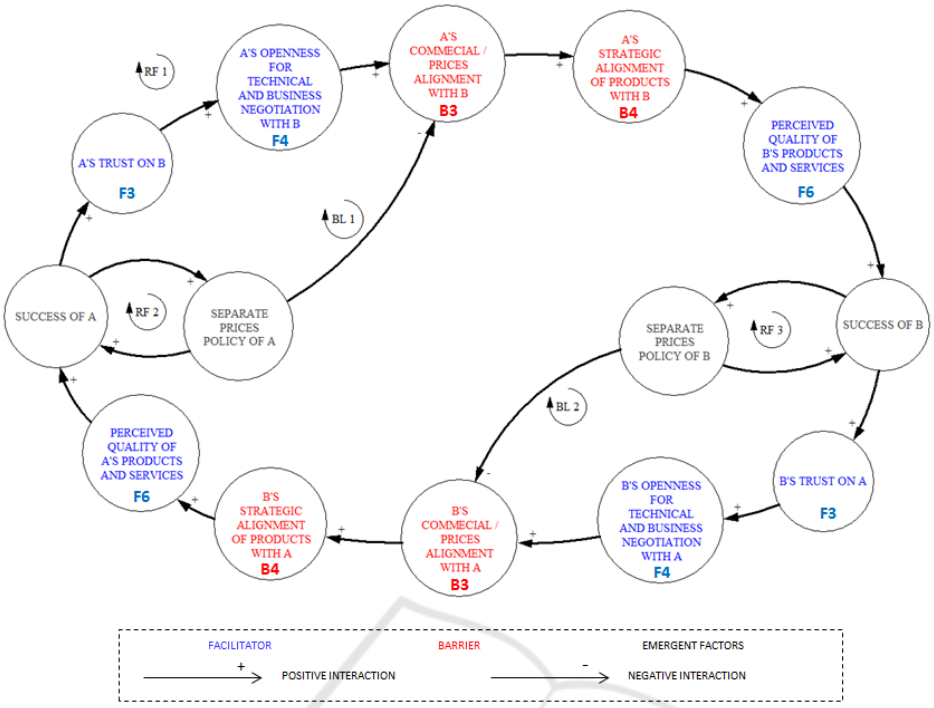

Figure 2: Accidental Adversaries system archetype representing interactions among barriers and facilitators.

support the healthy evolution of the software ecosystems. Hence, we address RQ3. The strategy S1 treats the barriers *poor strategic alignment of products* (B4), *overlap between features offered by the company and potential partners* (B5) and *limited authority over partner's development teams* (B7).

- **S1 – Partners Must Align Their Product Strategies to Sustain Ecosystem Evolution**

In a software ecosystem, a SME must try to align its business models with that of partners. If this firm has a power position, it may even succeed in putting partners onto its desired path. Hence, the company may lead others to want what it envisages (Yoffie and Kwak, 2006). In some cases, studied SMEs jointly analyse their commercial models (e.g. prices, sales process). However, this is an informal initiative of directors with closer relationships. Firms such as. Companies A and C are trying to promote the alignment of their product portfolios by sharing market intelligence with partners. This practice attaches partners to the ecosystem by fulfilling their business expectations. It also implicitly directs them Nowadays, software companies are expected to provide an overall view of their products evolution and decision-making about future product releases (Suomalainen et al., 2011). By following this trend and opening product roadmaps, partners embrace the mutual dependence that is required in an ecosystem. They start to give up the right of independently defining new features and share this privilege with others. Hence, studied firms shall enable ecosystem partners to influence changes in roadmaps regularly.

If the product roadmaps in the ecosystem are not correctly aligned, partners can have major problems, e.g. integration mismatches, solutions mutually competing and reduced co-innovation (Jansen et al. 2013). For example, there may be conflicts related to features functioning or removal of features due to supposed disuse by another system. To integrate their products properly, partners should make joint decisions regarding upcoming features.

Although a firm may gain the right to act as integration coordinator in a specific collaboration project, it does not exert sufficient control over partner teams. Hence, the coordinating company faces challenges to align the product releases of multiple vendors and treat integration conflicts. Partner companies can address this barrier by adopting a technical orchestration strategy that enables them to hold a new right: to access a partner development team. By gaining authority over each other's teams, a company can plan future product releases aligned with product evolutions from other ecosystem participants. The alignment of integrated solutions guarantees products' correct operation and reinforces companies' expertise in the ecosystem. Therefore, it increases the *perceived quality of*

products and services (F6), which ultimately fosters success of a company in the ecosystem, as perceived in the archetype in Figure 2. To treat the barriers *inefficiency to handle integration problems* (B1) and *poor strategic alignment of products* (B4), partner firms should also invest in the creation of a common software platform. This gives rise to the strategy S2.

- **S2 – SMEs Forming an Ecosystem can Jointly Develop a Software Platform**

To address the challenges involved in the integration of several products, partners can evolve their specific integration mechanisms towards a platform. This shared infrastructure may consist of services, tools and technologies that ecosystem members can use to enhance their performance (Iansiti and Levien, 2004). It enables the composition of features or services that can be accessed via common interfaces (Isckia and Lescop, 2009). The platform can enhance companies' expertise by supporting the development of valuable synergies and complementary innovations for partners and clients. SMEs shall start to evolve from a productisation to a platformisation approach (Artz et al., 2010), which is a strategic action to increase ecosystem health. Companies would then address resource constraints by attracting other suppliers to offer niche features, fostering a vibrant and potentially larger ecosystem around the platform.

Initially, the SMEs shall discuss how this platform will be offered and managed. Since the creation of this infrastructure represents an extra cost that is not funded by clients, partners could opt for a shared development and maintenance of the platform. Another option could be to evolve one firm's platform. However, negotiations and disputes around integration technologies may occur due to advantages that firms perceive in having platform ownership, e.g. become a keystone and control its influence in the ecosystem (Harland and Wüst, 2012). The software platform can enhance ecosystem productivity and robustness by enabling firms to build and integrate solutions more naturally.

Communication (F5) is a key factor to deal with the barrier *poor strategic alignment of products* (B4). In light of that, we propose the strategy S3:

- **S3 – Partners Must Develop Effective Communication Channels in the Ecosystem**

Our studies revealed that partners must improve their communication capabilities, as this process is still unstructured and immature. Their challenge resides in defining centralised and continuous communication channels. For instance, the manager of a joint project among the SMEs has great difficulty to interact with teams from partners. We observed that communication tends to be rich during the peak of product integration projects. Then, it gradually decreases and suddenly resumes as problems emerge. We also noticed other problems in the distribution of information among partners: integration and functional requirements that are not informed; artefacts that are not shared; problems that last to be solved; and new feature releases that are not reported to partners.

According to Jansen and colleagues (2009), one of the challenges in a software ecosystem is indeed to build common and efficient communication channels, which enable the orchestration of partners. To address this issue, Fricker (2010) recommends the use of traceability, audit trails and computer-supported collaborative work, for instance. These are means to obtain effective knowledge sharing and management among players in the ecosystem. By guaranteeing effective interchange of information (e.g. companies informing each other about product technological advances and upcoming features), partners can develop valuable complementary solutions. This is essential to strengthen ecosystem productivity as well as niche creation.

A final strategy proposed to the studied software ecosystems targets the balancing cycle *BL 1*, which may affect the positive cycle *RF 2* in the SD model presented in Figure 1. It means that the *weak commercial/prices alignment* (B3) will reinforce the *poor strategic alignment of products* (B4). To treat these critical factors, we elaborate the strategy S4:

- **S4 – Partners Must Develop and Agree on a Revenue Model for the Software Ecosystem**

Studied companies argued that some partnerships might not evolve due to mismatches in their commercial strategies. In particular, some companies believe that they can define prices independently of partners. This situation makes the integration of products hard or even unfeasible due to incompatible prices. It reveals a potential lack of commercial alignment (maybe due to a reluctance to perform commercial negotiations) that jeopardise the sustainable growth of partnerships.

A revenue model consists of one or more revenue streams, which define the way to get compensation from a good or service provided (Hyrynsalmi et al., 2012). For instance, in the case of software as a service, the client normally pays a subscription fee (Popp, 2011). In an ecosystem formed by big players such as Apple or Google, the keystone is responsible for defining the revenue model(s) adopted in the network, with which external agents must be aligned.

In the studied scenario, the software ecosystem partners can negotiate revenue models that are more suitable for their context. They must ensure a win-

win approach, with an egalitarian revenue model that do not cause partners migration to other networks, increasing software ecosystem robustness. In addition, such strategy shall fund innovation and subsidise new businesses, which can support niche creation by participants (Moore, 1993).

6 CONCLUSION

Companies participating in a software ecosystem co-create a collaborative network among their products. The success of software ecosystems involves managing a set of factors to foster the individual and collective health of the network. By understanding the positive and negative factors that affect partnerships, companies can derive strategies that leverage the facilitators while restraining the barriers. This paper presented a multiple case study of two software ecosystems. As contributions, we created SD models to analyse the interactions among facilitators and barriers. We also proposed a set of strategies to promote the evolution of the networks. Since these findings are applicable to other emergent ecosystems formed by SMEs, we invite researchers to assess our results and determine how closely their contexts match that of the case studies.

As future work, we plan to perform additional studies of similar software ecosystems. We believe it is possible to identify a set of factors and SD models that represent a pattern for such type of ecosystems, allowing a further generalisation of findings.

REFERENCES

Artz, P. et al., 2010. Productization: transforming from developing customer-specific software to product software. In *International Conference of Software Business*. Springer Berlin Heidelberg, pp. 90-102.

Coyne, I. T., 1997. *Sampling in qualitative research. Purposeful and theoretical sampling; merging or clear boundaries?*, Journal of advanced nursing 26 (3), pp. 623-630.

Cusumano, M. A., 2004. *The business of software: What every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad*, Simon and Schuster.

Fricker, S., 2010. Requirements value chains: Stakeholder management and requirements engineering in software ecosystems. In *International Working Conference on Requirements Engineering: Foundation for Software Quality*. Springer Berlin Heidelberg, pp. 60-66.

Harland, P. E., Wust, S., 2012. Strategic, brand and platform requirements for an interactive innovation process in business ecosystems. In *Int'l Conference on Engineering, Technology and Innovation*, pp. 1-9.

Hartigh, E., Tol, M., Visscher, W., 2006. The health measurement of a business ecosystem. In *European Chaos/Complexity in Organisations Network Conference*.

Hyrynsalmi, S., Suominen, A., Mäkilä, T., Järvi, A., Knuutila, T., 2012. Revenue models of application developers in android market ecosystem. In 3rd *Int'l Conference of Software Business*, pp. 209-222.

Iansiti, M., Levien, R., 2004. *Strategy as ecology*, Harvard Business Review 82 (3), pp. 68-81.

Isckia, T., Lescop, D. *Open innovation within business ecosystems: a tale from Amazon.com*, Communications & Strategies 74, pp. 37, 2009.

Jansen, S., Finkelstein, A., Brinkkemper, S., 2009. A sense of community: A research agenda for software ecosystems. In *31st International Conference on Software Engineering*, pp. 187-190.

Jansen, S., Peeters, S., Brinkkemper, S., 2013. Software Ecosystems: From Software Product Management to Software Platform Management. In *IW-LCSP@ ICSOB,* pp. 5-18.

Khalil, M. A. T. et al., 2011. A Study to Examine If Integration of Technology Acceptance Model's (TAM) Features Help in Building a Hybrid Digital Business Ecosystem Framework for Small and Medium Enterprises (SMEs). In *Frontiers of Information Technology*, pp. 161-166.

Manikas, K., Hansen, K. M., 2013. *Software ecosystems – a systematic literature review*, Journal of Systems and Software 86 (5), pp. 1294-1306.

Mohr, J., Spekman, R., 1994. *Characteristics of partnership success: partnership attributes, communication behavior, and conflict resolution techniques*, Strategic Management Journal 15 (2), pp. 135-152.

Moore, J. F., 1993. *Predators and prey: a new ecology of competition*, Harvard Business Review 71 (3), pp. 75-83.

Olsson, H. H., Bosch, J., 2016. Collaborative Innovation: A Model for Selecting the Optimal Ecosystem Innovation Strategy. In *42th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 206-213.

Popp, K.M., 2011. Hybrid revenue models of software companies and their relationship to hybrid business models. In *Third International Workshop on Software Ecosystems*, pp.77-88.

Popp, K. M., 2013. *Mergers and Acquisitions in the Software Industry: Foundations of due diligence*, BoD–Books on Demand.

Senge, P. M., Kurpius, D., 1993. *The fifth discipline: The art and practice of the learning organization*.

Suomalainen, T. et al., 2011. *Software product roadmapping in a volatile business environment*, Journal of Systems and Software 84 (6), pp. 958-975.

Weiblen, T., 2015. *Opening Up the Business Model: Business Model Innovation through Collaboration*, PhD Thesis. University of St. Gallen.

Yoffie, D. B., Kwak, M., 2006. *With friends like these: the art of managing complementors*, Harvard Business Review 84 (9), pp. 88-98.